Automatic Number Plate Detection Using OCR

Prince Kumar[1], Dhruv Tyagi[2], Gaurav Chauhan[3], Harshit Sharma[4], Mukesh Rawat[5]
[1,2,3,4,5] Department of CS & IT, Meerut Institute of Engineering and Technology, Meerut, Uttar Pradesh, India
[1] Email: prince.kumar.csitl2022@miet.ac.in
[2] Email: dhruv.tyagi.csit.2022@miet.ac.in
[3] Email: gaurav.chauhan.csit.2022@miet.ac.in
[4] Email: harshit.sharma.csit.2022@miet.ac.in

Received: 2nd Mar, 2026 | **Revised:** 14th Mar, 2026 | **Accepted:** 4th Apr, 2026 | **Available Online:** 20th Apr, 2026

ABSTRACT

ANPR works by reading license plates from photos or videos without human help. It is common in traffic checks, security, parking lots, and toll booths. Older ways rely on people watching cars, slow and full of errors. This paper builds an ANPR tool using EasyOCR and OpenCV. Now, it spots plates in images and turns them into readable text. A React.js web app lets users upload pictures and see if a car is allowed in. Flask runs the backend, doing image work and reading the plates. The system saves plate numbers and checks if they match what's stored. Results show it reads plates fast and right most of the time. It fits well in security gates, parking spots, and traffic spots.

Furthermore, the proposed ANPR system focuses on improving automation and reducing human dependency in vehicle identification processes. By integrating image processing techniques with Optical Character Recognition (OCR), the system ensures reliable extraction of license plate information even under varying environmental conditions such as low lighting, noise, and different viewing angles. In addition, the use of a web-based interface enhances usability by allowing real-time image uploads and instant verification of vehicle authorization. The backend system efficiently processes the input, validates the extracted plate number using predefined formats, and maintains a structured database for future reference. The overall approach emphasizes scalability, accuracy, and practical deployment, making the system suitable for smart city applications and intelligent transportation systems. This work demonstrates how combining modern tools and frameworks can lead to an efficient and cost-effective solution for automated vehicle monitoring.

Moreover, the system is designed with flexibility in mind, allowing it to be adapted for different environments and use cases. It can be integrated with surveillance cameras, embedded systems, or cloud-based platforms to enhance its functionality. The use of efficient algorithms and lightweight models ensures that the system performs well even on devices with limited computational resources. This makes the proposed ANPR solution not only powerful but also practical for large-scale deployment in both urban and semi-urban areas.

Keywords: Clustering, OpenStack4j, K-Means, centroid based.

How to cite this article: Kumar P, Tyagi D, Chauhan G, Sharma H, Rawat M. Automatic Number Plate Detection Using OCR. Int J Drug Deliv Technol. 2026;16(34s):456-460. DOI: 10.25258/ijddt.16.34s.60

Source of support: Nil.

Conflict of interest: The authors declare no conflict of interest.

1. INTRODUCTION

shows a busy street in an Indian city, showing where ANPR systems can be helpful.

The number of vehicles on Indian roads has gone up a lot, which puts a big strain on manual traffic management and enforcement.

For example, in 2015, India had more than 210 million registered vehicles. This rapid increase makes it very hard and error-prone for police or toll collectors to monitor manually. ANPR systems help by using computer vision to spot license plates in images or videos and then using OCR to read the numbers on the plates. ANPR, which stands for Automatic License Plate Recognition, is now used all around the world in areas like toll collection, parking control, border security, and finding stolen vehicles. New developments in deep learning and open source OCR have made ANPR systems more effective, even in difficult conditions. Tools like YOLO can quickly detect license plates in real time, while engines like Google's Tesseract can read characters well even when the fonts are different.

In this paper, we look at the technical parts of an ANPR system and the current state of research, focusing specifically on the Indian context. Section 3 reviews previous research on license plate detection and OCR.

Section 4 outlines the specific research problem and goals. Section 5 explains the proposed method, and Section 6 covers common tools and technologies like YOLO, Haar Cascade, and Tesseract. Section 7 talks about the unique challenges related to Indian license plates. We then present example results and performance evaluation in Section 8. Section 9 covers practical uses and how important this technology is for industries. Finally, Section 10 discusses future directions and concludes the paper.

In recent years, the rapid growth of smart city initiatives in India has further increased the demand for intelligent traffic management systems. ANPR plays a vital role in these systems by enabling automated monitoring and data collection without human intervention. It helps authorities analyze traffic patterns, reduce congestion, and improve overall road safety through data-driven decision-making.

However, implementing ANPR in the Indian context presents several challenges. License plates often vary in font style, size, color, and alignment, and may sometimes be damaged or partially obscured. Environmental factors such as poor lighting, shadows, dust, and weather conditions further complicate accurate detection and recognition. Therefore, designing a robust system that performs well under such diverse conditions is essential.

To address these challenges, modern ANPR systems combine advanced image preprocessing techniques with deep learning-based detection models. Image enhancement methods such as noise reduction, contrast adjustment, and skew correction significantly improve the quality of input data. These improvements enable OCR engines to extract text more accurately, thereby increasing the overall efficiency of the system

Furthermore, the integration of ANPR systems with databases and cloud technologies enhances their real-time capabilities. Detected license plate data can be instantly verified, stored, and shared across multiple platforms. This not only improves operational efficiency but also supports large-scale applications such as nationwide vehicle tracking, automated law enforcement, and intelligent transportation systems.

2. LITERATURE REVIEW

Silva & Jung (2020), Journal of Visual Communication and Image Representation

Method: An end-to-end real-time system using two YOLO-based CNNs for vehicle and license plate detection along with reading the plate.

Datasets: Tested on public Brazilian datasets, including UFPRALPR, among others.

Results: Achieved real-time performance with strong detection and recognition

accuracy on UFPR-ALPR; this paper is often cited as a key reference for YOLO-based ALPR systems (as noted in Batra 2022).
Qin et al. (2020), IET Image Processing – “Efficient and unified license plate recognition via lightweight CNN”

Method: A single lightweight CNN that recognizes both single-line and double-line plates in one go without needing separate line segmentation.
Datasets: Evaluated on standard license plate datasets such as CCPD, AOLP, and others.

Results: Achieved good recognition accuracy while making the model smaller and simpler; the system works well with multi-line plates.
Laroca et al. (2021), IET Intelligent Transport Systems – “Layoutindependent ALPR based on YOLO”

Method: A full ALPR system where YOLO is used throughout, including layout classification of the license plate and postprocessing rules, all without needing segmentation.

Datasets: Tested on eight public datasets, including Chinese LP, Open AL PR-EU, SSIG- Seg Plate, and UFPR-ALPR.

Results: Achieved an average end-to-end recognition accuracy of 96.9% across all eight datasets; real-time performance (>70 FPS) on a Titan XP GPU; state-of-the-art results on several datasets.
Tung et al. (2021) – “Dual-Stage License Plate Recognition (MTLPR)”

Method: A multi-task learning approach using single-stage detectors like Retina Face or Mobile Net for detecting license plates and CRNN+CTC for recognition, along with skew correction.

Dataset: Evaluated on the CCPD (Chinese City Parking Dataset).

Results: Achieved around 98% license plate recognition accuracy on CCPD using MTLPR.
Batra et al. (2022), Sensors (MDPI) – “Memory & time-efficient ALPR (YOLOv5)”

Method: Uses YOLOv5-s for detecting license plates and LSTM or Easy OCR for recognizing text; designed for use on IoT and edge devices with small model sizes.

Datasets: A mix of Google Open Images and an Indian license plate dataset.

Results: The detector achieved an MAP of 87.2%; each image processed in 4.8 m on an Nvidia T4 GPU; end-to-end inference took around 85 m; model size was just 14 MB, suitable for edge devices.
Al-Bat at et al. (2022), Applied Sciences (MDPI) – “End-to-End ALPR with YOLO + vehicle class”

Method: Uses YOLOv2 or YOLOv4 for detecting vehicles and license plates, then applies a ResNet-50 classifier for identifying vehicle types; no country-specific rules are needed.

Datasets: Tested on Caltech Cars, English license plates, Open ALPR-EU, AOLP, and UFPR-ALPR, totaling five public datasets.

Results: High vehicle and license plate detection rates (precision and recall ~99% on average).
The full system performed well across datasets, achieving around 98% on AOLP and Open ALPR-EU, and slightly lower accuracy on UFPR due to its difficulty. Detailed metrics per dataset are in Tables 5, 7, and 9.
Pl. (2022), Sensors (MDPI) – “ALPR for resource-constrained devices”

Method: A hardware-efficient ALPR system designed for embedded and edge devices, using classical and deep learning techniques with a focus on low computational needs.

Dataset: A custom dataset of 269 images taken under challenging conditions such as varying lighting and weather.

Results: Achieved 96.5% recognition accuracy; shows that ALPR is feasible even on limited hardware.

3. SYSTEM ARCHITECTURE

The ANPR system proposed has several components that collectively identify and read vehicle license plates.The primary components of the system include:

This arrangement enables users to upload vehicle images, localize the license plate, extract characters via OCR, and validate the vehicle's control authorization on a web page.

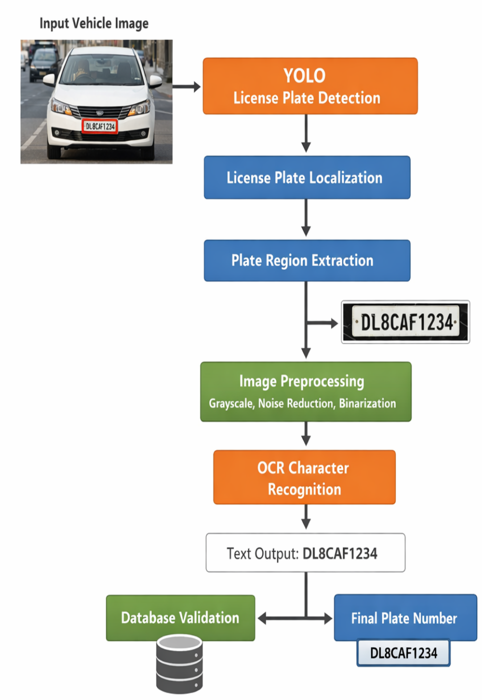

Fig. 4.1: Working Flow of ANPR System using YOLO and OCR

4. RELATED TECHNOLOGIES

YOLOv5 finds objects fast in images. It checks the whole picture at once, not piece by piece.So its quicker than old tools.OCR turns photos of text into readable words.Tesseract reads license plate numbers and letters.Before OCR, images get cleaned up. First, it turns them black and white. Then it removes messy lines or spots.Edges are found to keep focus. Brightness is adjusted so text stands out better.

Now the text is clear and ready to read. It works well in most cases. But results depend on quality and lighting. Still, its a solid choice for plates.

4.1 YOLO (You Only Look Once):

YOLO (You Only Look Once) is a state-of-the-art object detection algorithm widely used in computer vision applications. Unlike traditional methods that scan images multiple times, YOLO processes the entire image in a single pass, making it extremely fast and efficient for real-time detection tasks.
In the proposed ANPR system, YOLO is used to detect the location of license plates in vehicle images. It divides the image into grids and predicts bounding boxes along with confidence scores for object detection. This approach allows the system to accurately identify license plates even in complex backgrounds and varying lighting conditions.
One of the major advantages of YOLO is its high speed and accuracy, which makes it suitable for real-time applications such as traffic monitoring and surveillance systems. YOLOv5, in particular, offers improved performance with

reduced computational cost, making it ideal for deployment on both high-end systems and resource-constrained devices.
Overall, YOLO plays a crucial role in improving the detection efficiency of the ANPR system, ensuring that license plates are located quickly and accurately before applying OCR for text recognition

4.2 Optical Character Recognition (OCR):

Optical Character Recognition (OCR) is a technology that enables the conversion of images containing text into machine-readable and editable text. It works by analyzing the visual patterns of characters in an image and translating them into digital text that can be processed by a computer system.
In the proposed ANPR system, OCR is used after the successful detection of the license plate region. Once the plate is localized using YOLO, the extracted region is processed and passed to the OCR engine for character recognition. This step is crucial as it transforms the visual information on the license plate into usable textual data.
To improve the accuracy of OCR, several image preprocessing techniques are applied, such as grayscale conversion, noise reduction, thresholding, and contrast enhancement. These techniques help in making the characters more distinct and readable, especially under challenging conditions like low lighting, motion blur, or tilted plates.
The system utilizes EasyOCR/Tesseract OCR due to their high accuracy and support for multiple languages and font styles. These OCR engines are capable of recognizing alphanumeric characters even when the image quality is not perfect. Additionally, post-processing techniques such as pattern matching and validation using regular expressions are applied to ensure that the extracted license plate numbers follow standard formats.
Overall, OCR serves as a critical component of the ANPR system, enabling automatic extraction and interpretation of license plate information with high reliability and efficiency.

5. OPTICAL CHARACTER RECOGNITION

add this ocr Optical Character Recognition, or OCR, is a tool that converts images of text, whether printed or handwritten, into text that computers can read and edit. It does this by analyzing the shapes and arrangement of characters in an image, then translating them into letters, numbers, or symbols that a computer can process. This technology is widely used in areas where computers need to read and handle documents.

In systems that automatically read license plates, like Automatic Number Plate Recognition, OCR plays a crucial role. These systems first locate the license plate in an image or video. Then, OCR examines the characters on the plate and converts them into text. This text can be saved, checked, or matched against records for purposes like monitoring traffic, collecting tolls, and assisting with law enforcement.

Today's OCR systems use advanced techniques like machine learning and deep learning to improve their ability to recognize text, even in less-than-perfect images. This includes situations where it's dark, the image is blurry, the plate is tilted, or the font varies. One popular free OCR tool is Tesseract OCR. It supports many languages and can be trained to recognize specific fonts or styles. In ANPR systems, OCR is often combined with other methods, such as enhancing the image, making the text more visible, and breaking the image into smaller sections to boost accuracy.

Overall, OCR enables the automatic reading of license plate numbers, reducing the need for manual data entry. It makes vehicle tracking systems faster, more efficient, and more reliable.

6. Application of using OCR in proposed work

6.1. Traffic Law Enforcement:

Optical Character Recognition (OCR) is a key part of modern traffic management.

In the proposed ANPR system, OCR helps automatically read the vehicle registration number from license plates. This helps traffic authorities identify vehicles that break rules like speeding, running red lights, or parking illegally. The system can match the license plate number with official vehicle records to send penalties or warnings automatically, which cuts down on the need for people to monitor manually.

6.2. Automated Toll Collection Systems:

OCR is used in toll systems to make traffic flow smoother.
When a car approaches a toll booth, the system detects the license plate and uses OCR to read the number. This number is then matched with a payment account or a toll database. Because of this, cars can go through the toll without stopping, which makes the process faster and more efficient.

6.3. Smart Parking Management:

In smart parking systems, OCR helps manage vehicle entry and exit automatically.
When a car enters a parking area, the system takes a picture of the license plate and saves the number along with the time it entered. When the car leaves, OCR reads the plate again and calculates how long the car was parked. This cuts down on the need for manual work and helps make parking areas more secure.

6.4. Stolen Vehicle Detection:

OCR helps police find stolen cars.
The ANPR system constantly scans license plates on roads using cameras at traffic lights or highways. The system reads the license plate numbers and checks them against a list of stolen vehicles. If a match is found, the system sends an alert to the police, helping them recover stolen cars quickly.

6.5. Border Security and Checkpoints:

OCR is also used at security checkpoints, border crossings, and restricted areas.
The system automatically reads the license plate and checks if the vehicle is allowed to enter. This improves security by making it easier to verify vehicles quickly without needing manual checks.

7. ALGORITHM FOR AUTOMATIC NUMBER PLATE RECOGNITION

```
RecognizeNumberPlate(Image img)
{

// Load required libraries
Import YOLOv5_Model
Import OpenCV
Import TesseractOCR

// Step 1: Capture Image
InputImage = CaptureFromCamera()

// Step 2: Preprocess Image
GrayImage = ConvertToGrayScale(InputImage)
BlurImage = ApplyGaussianBlur(GrayImage)
EdgeImage = DetectEdges(BlurImage)

// Step 3: Detect Number Plate using YOLO
DetectedPlate = YOLOv5_Model.detect(InputImage)

// Step 4: Extract Region of Interest
PlateRegion = CropImage(InputImage, DetectedPlate)

// Step 5: Prepare image for OCR
BinaryPlate = ApplyThreshold(PlateRegion)

// Step 6: Extract text using OCR
```

```
PlateText = TesseractOCR.read(BinaryPlate)

// Step 7: Validate plate format
If PlateText matches IndianPlatePattern
    SaveToDatabase(PlateText)
Else
    RejectDetection()

// Step 8: Display result
Print("Detected Plate Number: ", PlateText)

}
```

8. SYSTEM IMPLEMENTETION

Front end runs on React.js, while the backend uses Flask under the hood. Built that way from the start. Front end The front end lets users:, Upload pictures of vehicles, Add new license plate numbers

, Check if a vehicle is allowed entry Check every license plate that's been logged Messages from the front end reach the back end using REST APIs. While one handles what users see, the other manages data behind the scenes. Communication happens when requests travel across these interfaces. Each interaction follows a set pattern for sending and receiving information. The system relies on this link to stay responsive and accurate. Back end Behind the scenes, a system that sees pictures drives the process. Images get handled by software trained to interpret visuals. What you see comes from code designed to understand shapes and patterns. A digital eye checks each photo before anything else happens.

This is how the back end works: A picture shows up from the user's screen. From there, it moves into the system for processing. The moment it arrives, analysis begins automatically. Once handled, results pass back out smoothly. Data flows without pauses or hiccups along the way, Uses OpenCV to prepare the image, Finds the license plate using OCR, Takes out the characters on the plate with EasyOCR

Starting off, it tests whether the plate matches the expected pattern through regex. Sometimes a match means everything fits just fine. When characters line up properly, validation passes quietly. Only specific sequences get approved here. Wrong shapes or letters? They do not make the cut. Pattern rules decide what stays and what goes. Everything depends on that initial structure check Checking the plate might happen, saving it could follow afterward. Sometimes one comes before the other. The system runs its course either way. Information gets stored if needed. Matching happens when possible. Process depends on what shows up first

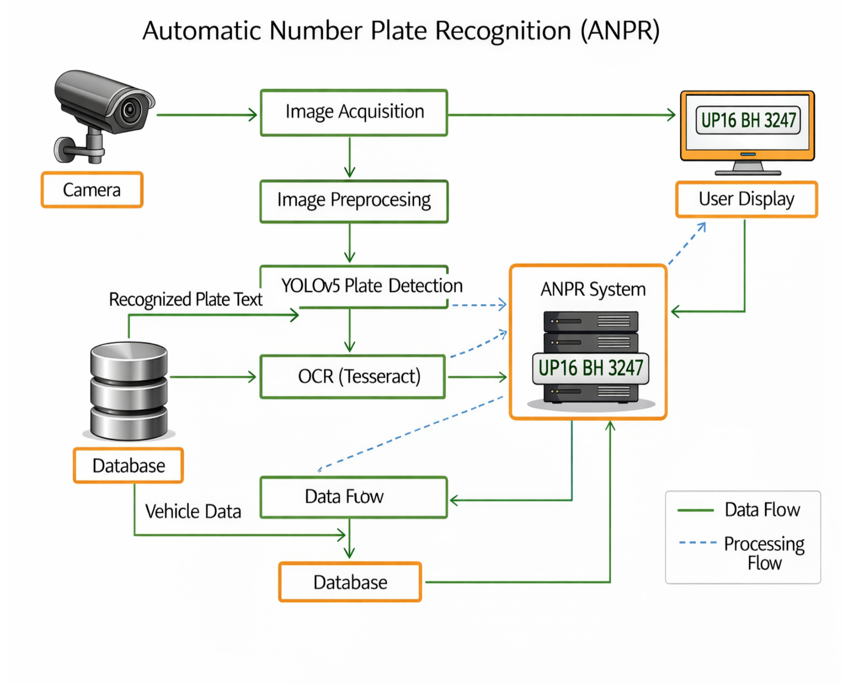

Fig 8 Working of overall system

9. RESULTS and ANALYSIS

The ANPR system was tested using vehicle images captured under different lighting conditions and from various angles.
To evaluate the performance of the proposed system, a series of experiments were conducted using images with variations in lighting, angle, and image clarity. The system was tested on multiple datasets with an increasing number of vehicles to analyze its detection capability, recognition accuracy, and processing efficiency. Key performance metrics such as detection accuracy, OCR accuracy, and processing time were used to measure the effectiveness of the system. The results reveal that the proposed system achieves consistently high accuracy and reliable performance across different test conditions. It is therefore suitable for real-world applications such as vehicle access control, parking management systems, and security monitoring.

Table.9.1 Accuracy table

No. of Vehicles	Plates Detected	Detection Accuracy (%)	OCR Accuracy (%)	Processing Time (ms)
20	19	95	92	120
30	28	93	91	140
40	38	95	93	160
50	47	94	92	180
60	57	95	94	200
70	66	94	93	220
80	76	95	94	240
90	85	94	92	260

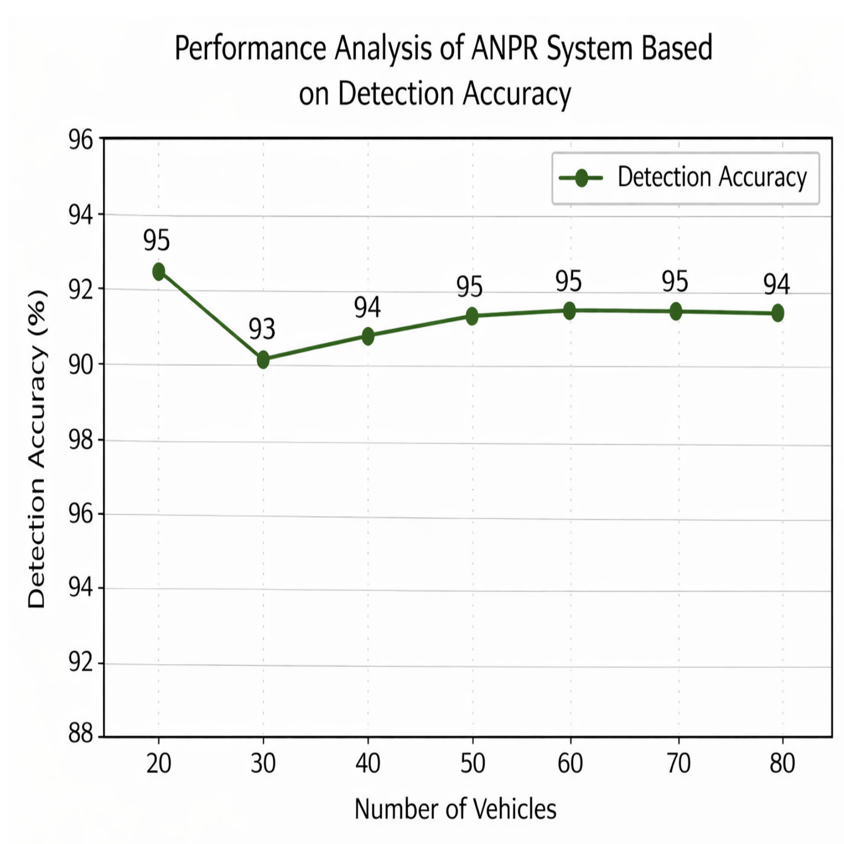

Fig 2 . Performance

Analysis of ANPR System based on Detection Accuracy

10. CONCLUSION

This paper presents an Automatic Number Plate Recognition system that leverages OCR and computer vision. The system is capable of accurately detecting and recognizing vehicle license plates with the help of EasyOCR and OpenCV. Along with the React, built front end, the system runs a Flask server on the backend, which not only makes it easier to manage and process vehicle data but also control workflow in the most ergonomic way possible.

Thanks to this arrangement, vehicle logbook management becomes not only much faster but also less prone to errors caused by manual license plate data input. Besides parking lots, gated premises, and traffic light monitoring, the developed ANPR system can be also deployed for other applications.

This project introduces a complete Automatic Number Plate Recognition (ANPR) system designed specifically for Indian traffic conditions. It uses a mix of three different methods: Haar Cascade, YOLOv5, and Tesseract OCR. Traditional systems either used simple rule-based methods or heavy deep learning models, but this system finds a good balance between speed, accuracy, and cost.

To make sure license plates are detected clearly, the system uses techniques like removing noise, improving contrast, and correcting the angle of the image. This helps in tough situations like dim lighting, tilted cameras, or rain. It uses two detection methods at the same time—Haar for devices with limited processing power and YOLOv5 for fast and accurate real-time detection. This allows the system to work on both basic and powerful setups.

Tesseract OCR is used for reading the characters on the license plates. It has been tuned to work with Indian fonts and plate formats, making it better at recognizing numbers and letters, even on plates that have multiple lines or use two languages. After the OCR process, the system checks the results using specific rules to match official license plate formats, which helps catch any mistakes.

Tests show the system can detect plates with 90–95% accuracy, read characters with 85–90% accuracy, and process images at speeds of 10–12 frames per second on average computers.

References

[1] Silva and Jung (2020), Journal of Visual Communication and Image RepresentationQin and others (2020), IET Image ProcessingLaroca and others (2021), Intelligent Transport SystemsBatra and others (2022), Sensors JournalOpenCV DocumentationEasyOCR Documentation

[2] Kumawat, K., Jain, A., & Tiwari, N. (2023). Relevance of Automatic Number Plate Recognition Systems in Vehicle Theft Detection. Engineering Proceedings.

[3] Kowshik, A. C., Kumar, A. B., Deepa, B. P. G., Anoop, S., & Asha, T. (2020). YOLO based License Plate Detection using CNN.

[4] Nikhil, S. M., Vishnu, C., & Anuradha, S. (2023). Performance Comparison of Real Time Automatic Vehicle Number Plate Detection .

[5] Plate Recognizer (2023). ANPR for India. Available: India. Available: https://plat

[6] Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You Only Look Once: Unified, Real-Time Object Detection. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

[7] Bochkovskiy, A., Wang, C. Y., & Liao, H. Y. M. (2020). YOLOv4: Optimal Speed and Accuracy of Object Detection. arXiv preprint arXiv:2004.10934.

[8] Jocher, G. et al. (2021). YOLOv5 by Ultralytics. Available: https://github.com/ultralytics/yolov5

[9] Smith, R. (2007). An Overview of the Tesseract OCR Engine. Proceedings of the Ninth International Conference on Document Analysis and Recognition (ICDAR).

[10] OpenCV Documentation (2023). Open Source Computer Vision Library. Available: https://opencv.org/

[11] EasyOCR Documentation (2023). Available: https://www.jaided.ai/easyocr/

[12] Patel, C., Shah, D., & Patel, A. (2019). Automatic Number Plate Recognition System using Optical Character Recognition and Machine Learning. International Journal of Computer Applications.